CLAIMS

What is claimed is:

1. A method comprising:

performing security authentication of a content driver in order to verify an identity of the content driver as a secure content driver;

receiving an encrypted content stream from the secure content driver;

performing integrity authentication of a run-time image of the secure content driver; and

while integrity authentication of the secure content driver is verified, streaming decrypted content to the secure content driver to enable playback of the decrypted content to a user.

2. The method of claim 1, wherein performing security authentication further comprises:

locating authorization information of the secure content driver;

decrypting the authorization information received from the secure content driver;

authenticating an identity of the secure content driver based on the decrypted authorization information; and

once the identity of the secure content driver is authenticated, providing the secure content driver with access to a callback function, such that access to the callback function enables the secure content driver to receive the decrypted content stream.

3. The method of claim 2, wherein authenticating the identity further comprises:

calculating a hash value of a static image of the secure content driver prior to loading the secure content driver into memory;

selecting a stored digital signature of the static image;

decrypting the stored digital signature to retrieve a pre-calculated hash value of the secure content driver;

comparing the pre-calculated hash value with the calculated hash value; and

when the calculated hash value matches the pre-calculated hash value of the secure content driver, notifying the secure content driver of successful security authentication.

4. The method of claim 1, wherein performing security authentication further comprises:

once security authentication of the content driver is established, determining a run-time at memory location of the secure content driver; and

establishing a function entry point for receiving the stream of encrypted content from the secure content driver.

5. The method of claim 1, further comprising:

receiving a content decryption key in order to enable decryption of encrypted content streams received from the secure content driver;

receiving a digital signature of a static image of the secure content driver; and

receiving a digital signature of a run-time image of the secure content driver.

6. The method of claim 1, wherein performing integrity authentication further comprises:

decrypting the encrypted content stream received from the secure content driver;

while decrypting the received encrypted content stream, performing a hash value calculation of code segments that perform functionality of the secure content driver while loaded in memory;

selecting a stored digital signature of a run-time image of the secure content driver;

decrypting the digital signature to reveal a run-time hash value;

comparing the computed hash value with the run-time hash value of the secure content driver; and

while the calculated hash value matches the run-time hash value of the secure content driver, repeating the decryption, the performing, the selecting and the comparing until decryption of the received encrypted content stream is complete.

7. A method comprising:

establishing security authentication from a content decryption component, such that a content driver is verified as a secure content driver;

when establishment of security authentication is successful, receiving access to a callback function in order to receive clear, decrypted content streams from the content decryption component;

receiving a stream of encrypted content;

streaming the encrypted content to the content decryption component; and

when security authentication is successfully established, receiving clear, decrypted content from the content decryption component via the received callback function.

8. The method of claim 7, wherein establishing security verification further comprises:

receiving a request for authorization information from the content decryption component;

transmitting the requested authorization information to the content decryption component; and

when security authentication is successfully established, receiving notification of successful security authentication from the content decryption component, such that the content driver is established as the secure content driver.

9. The method of claim 7, wherein establishing security authentication further comprises:

once security authentication is established, providing content decryption component with a memory location wherein the secure content driver is loaded at run-time; and

providing the content decryption component with a function entry point for receiving the stream of encrypted content.

10. The method of claim 7, wherein receiving encrypted content further comprises:

receiving encrypted content from a content source reader; and

receiving a direction from a content driver to stream the encrypted content to the content decryption component.

11. A computer readable storage medium including program instruction that directs a computer to function in a specified manner when executed by a processor, the program instructions comprising:

performing security authentication of a content driver in order to verify an identity of the content driver as a secure content driver;

receiving an encrypted content stream from the secure content driver;

performing integrity authentication of a run-time image of the secure content driver; and

while integrity authentication of the secure content driver is verified, streaming decrypted content to the secure content driver to enable playback of the decrypted content to a user.

12. The computer readable storage medium of claim 11, wherein performing security authentication further comprises:

locating authorization information of the secure content driver;

decrypting the authorization information received from the secure content driver;

authenticating an identity of the secure content driver based on the decrypted authorization information; and

once the identity of the secure content driver is authenticated, providing the secure content driver with access to a callback function, such that access to the callback function enables the secure content driver to receive the decrypted content stream.

13. The computer readable storage medium of claim 12, wherein authenticating the identity further comprises:

calculating a hash value of a static image of the secure content driver prior to loading the secure content driver into memory;

selecting a stored digital signature of the static image;

decrypting the stored digital signature to retrieve a pre-calculated hash value of the secure content driver;

comparing the pre-calculated hash value with the calculated hash value; and

when the calculated hash value matches the pre-calculated hash value of the secure content driver, notifying the secure content driver of successful security authentication.

14. The computer readable storage medium of claim 11, wherein performing security authentication further comprises:

once security authentication of the content driver is established, determining a run-time at memory location of the secure content driver; and

establishing a function entry point for receiving the stream of encrypted content from the secure content driver.

15. The computer readable storage medium of claim 11, further comprising:

receiving a content decryption key in order to enable decryption of encrypted content streams received from the secure content driver;

receiving a digital signature of a static image of the secure content driver; and

receiving a digital signature of a run-time image of the secure content driver.

16. The computer readable storage medium of claim 11, wherein performing integrity authentication further comprises:

decrypting the encrypted content stream received from the secure content driver;

while decrypting the received encrypted content stream, performing a hash value calculation of code segments that perform functionality of the secure content driver while loaded in memory;

selecting a stored digital signature of a run-time image of the secure content driver;

decrypting the digital signature to reveal a run-time hash value;

comparing the computed hash value with the run-time hash value of the secure content driver; and

while the calculated hash value matches the run-time hash value of the secure content driver, repeating the decryption, the performing, the selecting and the comparing until decryption of the received encrypted content stream is complete.

17. A computer readable storage medium including program instruction that directs a computer to function in a specified manner when executed by a processor, the program instructions comprising:

establishing security authentication from a content decryption component, such that a content driver is verified as a secure content driver;

when establishment of security authentication is successful, receiving access to a callback function in order to receive clear, decrypted content streams from the content decryption component;

receiving a stream of encrypted content;

streaming the encrypted content to the content decryption component; and

when security authentication is successfully established, receiving clear, decrypted content from the content decryption component via the received callback function.

18. The computer readable storage medium of claim 17, wherein establishing security verification further comprises:

receiving a request for authorization information from the content decryption component;

transmitting the requested authorization information to the content decryption component; and

when security authentication is successfully established, receiving notification of successful security authentication from the content decryption component, such that the content driver is established as the secure content driver.

19. The computer readable storage medium of claim 17, wherein establishing security authentication further comprises:

once security authentication is established, providing content decryption component with a memory location wherein the secure content driver is loaded at run-time; and

providing the content decryption component with a function entry point for receiving the stream of encrypted content.

20. The computer readable storage medium of claim 17, wherein receiving encrypted content further comprises:

receiving encrypted content from a content source reader; and

receiving a direction from a content driver to stream the encrypted content to the content decryption component.

21. An apparatus, comprising:

a processor having circuitry to execute instructions;

a content play-back interface coupled to the processor, the content play-back interface to receive encrypted content, and to enable play-back of the received encrypted content to a user; and

a storage device coupled to the processor, having sequences of instructions stored therein, which when executed by the processor cause the processor to:

perform security authentication of a content driver in order to verify an identity of the content driver as a secure content driver,

receive an encrypted content stream from the secure content driver,

perform integrity authentication of a run-time image of the secure content driver, and

while integrity authentication of the secure content driver is verified, stream decrypted content to the secure content driver to enable playback of the decrypted content to a user.

22. The apparatus of claim 21, wherein the instruction to perform security authentication further comprises the processor to:

locate authorization information of the secure content driver,

decrypt the authorization information received from the secure content driver,

authenticate an identity of the secure content driver based on the decrypted authorization information, and

once the identity of the secure content driver is authenticated, provide the secure content driver with access to a callback function, such that access to the callback function enables the secure content driver to receive the decrypted content stream.

23. The apparatus of claim 22, wherein the instruction to perform security authentication further comprises the processor to:

calculate a hash value of a static image of the secure content driver prior to loading the secure content driver into memory,

select a stored digital signature of the static image,

decrypt the digital signature to retrieve a pre-calculated hash value of the secure content driver,

compare the pre-calculated hash value with the calculated hash value, and

when the calculated hash value matches the pre-calculated hash value of the secure content driver, notify the secure content driver of successful security authentication.

24. The apparatus of claim 21, wherein the instruction to perform security authentication further comprises the processor to:

once security authentication of the content driver is established, determine a run-time at memory location of the secure content driver, and

establish a function entry point for receiving the stream of encrypted content from the secure content driver.

25. The apparatus of claim 21, wherein the processor is further caused to:

receive a content decryption key in order to enable decryption of encrypted content streams received from the secure content driver,

receive a digital signature of a static image of the secure content driver, and

receive a digital signature of a run-time image of the secure content driver.

26. The apparatus of claim 21, wherein the instruction to perform security authentication further comprises the processor to:

decrypt the encrypted content stream received from the secure content driver,

while decrypting the received encrypted content stream, perform a hash value calculation of code segments that perform functionality of the secure content driver while loaded in memory,

select a stored digital signature of a run-time image of the secure content driver,

decrypt the digital signature to reveal a run-time hash value,

compare the computed hash value with the run-time hash value of the secure content driver, and

while the calculated hash value matches the run-time hash value of the secure content driver, repeat the decryption, the performing, the selecting and the comparing until decryption of the received encrypted content stream is complete.

27. The apparatus of claim 21, wherein the processor is further caused to:

establish security authentication from a content decryption component, such that a content driver is verified as a secure content driver,

when establishment of security authentication is successful, receive access to a callback function in order to receive clear, decrypted content streams from the content decryption component,

receive a stream of encrypted content,

stream the encrypted content to the content decryption component, and

when security authentication is successfully established, receive clear, decrypted content from the content decryption component via the received callback function.

28. The apparatus of claim 21, wherein the instruction to establish security verification further comprises the processor to:

receive a request for authorization information from the content decryption component,

transmit the requested authorization information to the content decryption component, and

when security authentication is successfully established, receive notification of successful security authentication from the content decryption component, such that the content driver is established as the secure content driver.

29. The apparatus of claim 21, wherein the instruction to establish security authentication further comprises the processor to:

once security authentication is established, provide content decryption component with a memory location wherein the secure content driver is loaded at run-time, and

provide the content decryption component with a function entry point for receiving the stream of encrypted content.

30. The apparatus of claim 21, wherein the instruction to receive encrypted content further comprises the processor to:

receive encrypted content from a content source reader, and

receive a direction from a content driver to stream the encrypted content to the content decryption component.